Predicting Shot-Creating Passes in Soccer Using Transformer Models: A Sequence-Based Analysis of Leicester City's 2015–16 Season

Kejian Zhu
University of Michigan
zhukj@umich.edu

Abstract

Understanding how danger emerges in soccer requires modeling the sequential structure of match events rather than treating actions in isolation. This paper investigates whether a Transformer encoder can identify pass sequences that precede dangerous outcomes, defined as shots taken within ten events following a pass with expected goals (xG) of at least 0.05. Using complete event data from Leicester City's 2015–16 Premier League campaign, the model processes six-pass windows and predicts whether the final pass will lead to a shot. A logistic regression classifier serves as a baseline and is evaluated on the same sequence-level test set. The logistic model achieves an AUC of 0.887 and AP of 0.308, while the Transformer achieves an AUC of 0.887 and a substantially higher AP of 0.410. Although the global discrimination (AUC) is similar, the Transformer exhibits significantly improved ranking of rare dangerous events, suggesting that sequence context and player-interaction embeddings meaningfully enhance predictive precision. Interpretability analyses demonstrate that the model captures tactical patterns consistent with Leicester City's transitional attacking style. These findings highlight the value of sequence-based deep learning approaches for soccer event prediction.

1 Introduction

Soccer is a sequential, context-dependent game in which the value of each action depends strongly on the preceding flow of play. A single pass can be safe or extremely dangerous depending on prior movements, field position, and tactical structure. Traditional soccer analytics often models actions in isolation, focusing on single events such as shots, passes, or duels, and assigning them values through possession-value frameworks like Expected Threat (xT) (1). While these approaches have been successful in quantifying the contribution of individual actions, they typically assume that threat is primarily a function of location and immediate context rather than longer attacking sequences.

The 2015–16 Leicester City team provides a compelling case study for sequence-based modeling. Leicester's title-winning campaign was built on rapid transitions, vertical attacks, and coordinated interplay among players such as Jamie Vardy, Riyad Mahrez, N'Golo Kanté, Marc Albrighton, and Christian Fuchs. Many of their most dangerous situations emerged not from isolated high-value passes but from multi-pass sequences that stretched the opposition and created space before the final action. Capturing these dynamics requires models that can reason about temporal context and player interaction patterns across multiple events.

This paper investigates whether a Transformer-based sequence model can predict dangerous passes more effectively than a simple logistic regression baseline. Using event-level data from Leicester City's 2015–16 Premier League matches, we construct short sequences of six consecutive Leicester passes and predict whether the final pass in the window is followed by a Leicester shot within the next ten events, with expected goals (xG) at least 0.05. The Transformer processes the entire six-pass

sequence, while the logistic regression baseline uses only the numeric features of the final pass. Both models are trained and evaluated on the same sequence-level train, validation, and test splits, allowing for a fair comparison.

The main contributions of this work are threefold. First, we introduce a sequence-based prediction task focused on identifying shot-creating passes in a real-world season for a specific team. Second, we implement and evaluate a Transformer architecture that combines spatial features, temporal ordering, and player identity embeddings, and we compare it against a logistic regression baseline trained on the same sequence endpoints. Third, we provide an interpretability analysis of the learned model, illustrating how it captures tactical patterns consistent with Leicester City's transitional attacking style.

2 Related Work

Soccer analytics has increasingly turned to event and tracking data to quantify performance, evaluate player contributions, and model match dynamics. Possession-value models such as Expected Threat (xT) assign values to locations on the pitch based on the probability that a possession will eventually result in a goal (1; 2). Recent extensions like Dynamic Expected Threat (DxT) incorporate player positions to make action evaluation more context-sensitive (3). These frameworks typically operate at the level of single actions, updating threat values as the ball moves, but do not explicitly model sequences of actions as inputs to a predictive model.

Several studies have explored machine learning approaches for predicting outcomes of possessions or phases of play using event data. Stival et al. use a machine learning pipeline to predict whether a team successfully enters the attacking third within a few seconds of gaining possession (4). Other work has used event data from providers such as StatsBomb and Wyscout to model match outcomes or to quantify player impact across matches (5; 6). These approaches focus on higher-level outcomes (e.g., match result, attack success) rather than directly modeling the probability that a specific pass will lead to a shot.

Deep learning methods and sequence models have recently been applied to soccer event streams. Simpson et al. introduced Seq2Event, a framework that uses recurrent or Transformer encoders to predict the next event and its attributes from sequences of match events (7). Transformer-based spatio-temporal point process models have been proposed to jointly predict the timing, location, and type of future events (8). Other recent work has used Transformers to assess player performance and to provide a versatile framework for analyzing soccer actions (9). Beyond soccer, sequence models, including Transformers, have been successfully applied to event prediction in other sports and domains (10; 11).

Passing sequences themselves have been studied using network and pattern-mining approaches. McCarthy et al. analyze passing sequences to infer the style of play and relate them to goal-scoring opportunities (12). Clustering and motif-based methods have been used to identify line-breaking passes and to uncover tactical structures that create or reduce threat (13). These approaches often rely on handcrafted features or unsupervised clustering, whereas our work trains a supervised model end-to-end to predict whether a sequence will culminate in a dangerous shot.

Our study is closest in spirit to Seq2Event and Transformer-based soccer models, but differs in its focus on a specific, interpretable outcome: whether a six-pass sequence leads to a shot within the next ten events. By restricting the analysis to a single season for a single team, we can connect the learned patterns to known tactical structures rather than aiming for a fully general model across leagues.

3 Method

3.1 Data and Preprocessing

We use event-level data from StatsBomb Open Data for the 2015–16 Premier League season (14). All matches involving Leicester City are retrieved via the StatsBombPy API. For each match, we access the full sequence of events, including passes, shots, duels, and other actions, and attach the match identifier.

The preprocessing pipeline focuses on Leicester City passes. We first filter the event log to retain only Leicester events and then further restrict to events labeled as passes. For each pass, we extract the starting and ending locations on the pitch and compute engineered spatial features, including normalized coordinates on a 120-by-80 pitch, pass length, and pass angle. We convert boolean flags such as under pressure and counterpress into binary indicators and encode categorical variables, including pass height, body part, pass type, and play pattern, using numeric category codes. Player and pass recipient identities are encoded as integer indices, with a special index reserved for missing recipients to ensure that all passes have a valid receiver code.

To associate each pass with a danger label, we sort all events in each match by period, minute, second, and internal index, and assign an event index. A pass is labeled as dangerous if, within the next ten events in the same match, there exists a Leicester shot with expected goals (xG) at least 0.05. This threshold filters out very low probability attempts while retaining shots that represent meaningful attacking outcomes. The label is binary and does not distinguish between different shot qualities beyond the threshold.

Finally, we drop passes with missing key features or invalid player identifiers and standardize numeric features using a global StandardScaler. This cleaned dataframe provides the basis for both the logistic regression baseline and the sequence construction.

3.2 Sequence Construction

The task is to predict whether the final pass in a short attacking sequence will lead to a shot in the near future. To create sequences, we group passes by match and sort them chronologically within each match. For each match, we slide a fixed-length window of six consecutive Leicester passes. For a given window, the model input consists of the numeric features of each pass, the passer and receiver indices, and the positional index (0 to 5) in the sequence. The target label is the danger label of the final pass in the window.

Only matches with at least six Leicester passes contribute sequences. For each valid window, we obtain a tensor of shape $[L, F]$ for numeric features, where $L = 6$ and F is the number of numeric features, as well as vectors of passer and receiver indices of length L. The corresponding label is a scalar indicating whether the final pass leads to a dangerous shot.

We construct the full set of sequences and then split them into training, validation, and test subsets using stratified sampling on the labels. This ensures that each split maintains a similar proportion of dangerous and non-dangerous sequences, which is important because dangerous sequences are relatively rare.

3.3 Models

Logistic Regression Baseline. The baseline model is a logistic regression classifier trained on the same sequence dataset but using only the numeric features of the final pass in each six-pass window. For a given sequence, we extract the feature vector of the last pass and use it as input to the logistic regression. This ensures that the logistic model operates on the same samples and labels as the Transformer, making the comparison fair. The classifier is trained with class-balanced weights and a maximum of 1000 iterations.

Transformer Sequence Model. The main model is a Transformer encoder that processes the full six-pass sequence. Numeric features for each pass are projected into a latent space via a linear layer. Passer and receiver indices are mapped to learned embeddings, which are also projected to the same latent dimension. Positional embeddings indicate the temporal order of passes within the window. The sum of the projected numeric features, passer embeddings, receiver embeddings, and positional embeddings yields the input to the Transformer encoder.

The encoder comprises several layers of multi-head self-attention and feedforward sublayers, with layer normalization and dropout for regularization. The output is a contextualized representation for each pass in the sequence. We extract the hidden state corresponding to the final pass and feed it into a small feedforward network that outputs a scalar logit. The model is trained using binary cross-entropy with logits, with an additional positive class weight to address label imbalance. The

Adam optimizer is used with a fixed learning rate, and the best model is selected based on validation AUC.

4 Experiment

4.1 Experimental Setup

After constructing all valid sequences, we perform a stratified split into training, validation, and test subsets. The logistic regression baseline and the Transformer are both trained using the training set, with hyperparameters tuned informally using the validation set. Final performance is reported on the held-out test set only.

We evaluate both models using two metrics: the area under the ROC curve (AUC) and average precision (AP). AUC measures the overall ability of the model to rank dangerous sequences above non-dangerous ones across all thresholds. AP summarizes the precision–recall trade-off and is particularly informative in imbalanced settings where positive examples are rare. In the context of football analytics, AP is arguably more aligned with practical use cases, since analysts are often interested in the reliability of the top-ranked predicted dangerous passes.

4.2 Results

On the sequence-level test set, the logistic regression baseline achieves an AUC of 0.887 and an AP of 0.308. These results show that the numeric features of the final pass alone contain substantial information about whether a shot will occur in the near future. In particular, spatial location, pass length, angle, and contextual attributes such as pass type and play pattern already allow a linear model to distinguish many dangerous from non-dangerous actions.

The Transformer model, trained on the full six-pass sequence with player and receiver embeddings, obtains an AUC of approximately 0.88 and an AP of 0.41 on the same test set. The AUC is very similar to that of the logistic regression baseline, suggesting that both models achieve comparable global discrimination when ranking all sequences. However, the AP improves noticeably from 0.308 to around 0.41. This indicates that the Transformer is better at concentrating true dangerous sequences near the top of its ranking, which is particularly valuable when analysts focus on the most dangerous predicted events.

The fact that AUC remains similar while AP improves is consistent with the idea that sequences add subtle but important information that is most useful in the high-risk region. The logistic model is competitive when evaluating all thresholds, but the Transformer provides a more refined ranking of the most threatening passes.

4.3 Interpretability and Tactical Insights

Although the primary goal is predictive performance, the sequence model also yields insights into Leicester City's attacking structure. By examining the learned representations and scores, we observe that dangerous sequences often involve repeated interactions among well-known attacking players. Passer–receiver pairs such as Albrighton to Huth or Morgan, Fuchs to central defenders or Mahrez, and combinations involving Kanté, Mahrez, Okazaki, and Vardy appear frequently in high-probability sequences. These patterns align with established descriptions of Leicester's transitional style, in which the team built attacks via wide distribution from full-backs and deep players before accelerating through Mahrez and Vardy.

Sequence-level importance scores, computed by comparing intermediate Transformer hidden states to the representation of the final pass, suggest that early passes in a six-pass window often receive substantial weight. This supports the notion that danger is not solely determined by the final action, but by how earlier passes shape the defensive structure and field position. Spatial plots of final pass locations, colored by predicted probability, show dense clusters in wide and half-space regions near the penalty area, consistent with cutback and crossing zones that are known to be high-value in practice.

5 Conclusion and Discussion

This paper presented a sequence-based approach to predicting shot-creating passes in soccer using a Transformer model applied to Leicester City's 2015–16 Premier League season. By constructing six-pass windows and predicting whether the final pass leads to a shot within the next ten events, we framed the problem as a supervised sequence classification task. A logistic regression baseline trained on the same sequence endpoints achieved strong AUC but relatively modest average precision. The Transformer matched the baseline in AUC while significantly improving AP, indicating that sequential context and player-interaction embeddings help the model prioritize truly dangerous sequences near the top of the ranking.

From a methodological standpoint, the results show that even in a relatively small, team-specific dataset, deep sequence models can provide meaningful gains over simple baselines when evaluated with ranking-focused metrics. At the same time, the strong performance of the logistic regression model cautions against overstating the advantage of deep models; much of the signal remains encoded in spatial and contextual features of the final pass itself. The primary value of the Transformer appears in its ability to exploit temporal dependencies and repeated interaction patterns that are not fully captured by single-pass features.

For soccer practitioners, this framework suggests a way to identify and quantify multi-pass sequences that are likely to lead to shots, with clear tactical interpretations. Analysts could apply similar models to other teams or seasons to compare attacking structures, evaluate how player departures or signings change dangerous combinations, or simulate how different sequence patterns affect the likelihood of shot creation.

Future work could extend this study in several directions. Incorporating tracking data would allow the model to account for defender positions and off-ball movements, likely improving both predictive accuracy and tactical interpretability. Training on multi-team or multi-league datasets would test the generality of the learned patterns and potentially support transfer learning applications. Finally, integrating counterfactual or reinforcement learning frameworks could enable simulations in which alternative passing choices within a sequence are evaluated in terms of their effect on the probability of generating a dangerous shot.

Overall, this case study illustrates how modern sequence models, and Transformers in particular, can be adapted to soccer analytics to move beyond single-action evaluation and toward a richer understanding of how coordinated sequences of events create threat.

The code is available at: https://github.com/zhukejianbot/LeicesterPass

References

[1] Karun Singh. Introducing Expected Threat (xT): Modelling team behaviour in possession to gain a deeper understanding of buildup play. Blog post, 2018. Available at `https://karun.in/blog/expected-threat.html`.

[2] Footballytics. Expected Threat (xT): the best offensive metric so far. Blog post, 2022.

[3] Karim Hassani et al. Dynamic Expected Threat (DxT) Model: Addressing the Deficit of Realism in Football Action Evaluation. *Applied Sciences*, 15(8), 2025.

[4] Leandro Stival et al. Using machine learning pipeline to predict entry into the attack zone in soccer. *PLOS ONE*, 18(2): e0265372, 2023.

[5] Peter Hassard et al. Predicting football match outcomes using event data and machine learning algorithms. Technical report, 2024.

[6] Tahmeed Tureen and Sigrid Olthof. Estimated Player Impact (EPI): Quantifying the Effects of Individual Players on Football Actions. StatsBomb Conference Paper, 2022.

[7] Ian Simpson et al. Seq2Event: Learning the Language of Soccer Using Transformer-based Event Representations. In *Proceedings of KDD*, 2022.

[8] Jing Li et al. Transformer-based Neural Marked Spatio-Temporal Point Process for Football Events. *Applied Intelligence*, 2025.

[9] Andrey Rovshitz et al. SoccerTransformer: A Transformer-based Framework for Versatile Analysis of Soccer Players. In *Proceedings of MLSA*, 2024.

[10] Jian Gao et al. Predicting Sport Event Outcomes Using Deep Learning. *Journal of Sports Analytics*, 2025.

[11] Kevin McGuigan et al. A hard-hitting evaluation of deep learning approaches to predicting tackle events in the NFL. *Vision Science Letters*, 2024.

[12] Conor McCarthy et al. Analyzing Passing Sequences for the Prediction of Goal-Scoring Opportunities. In *Workshop on Machine Learning and Data Mining for Sports Analytics (MLSA)*, 2022.

[13] Marc Lamberts et al. Uncovering Tactical Line-Breaking Passes with Clustering. arXiv preprint, 2025.

[14] StatsBomb. StatsBomb Open Data. GitHub repository, available at `https://github.com/statsbomb/open-data`.